Old Globe White Paper

Content

Client History

Our client is The Old Globe Theatre a community theater in San Diego - California. The theater was built in early 1935 for the presentation of abridged versions of Shakespeare's plays as part of the California Pacific International Exposition.

The theater has seen many remold, renovation and new constructions since the inception. The development process is almost continuous. Taking care of accounts and stocks are primary concern of the administration department.

Existing system overview

To take care of account and stocks they have developed an application using Microsoft Access database. The system was believed to be developed around 1995. Different departments are using this application since then. They start getting delays in data entry and report generation due to Access data management limitation. This leads them to create new system with modern approach.

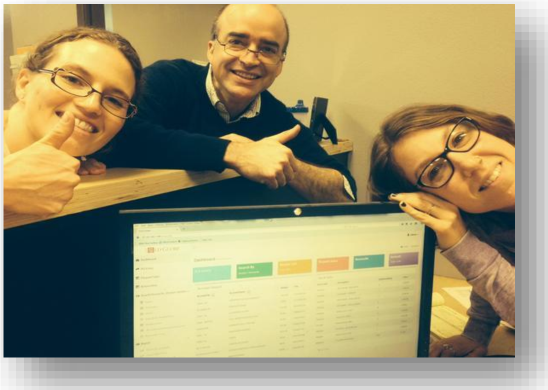
New requirement overview

The new system we developed after deeply analyzing their requirements and studying their various use-cases we developed a web application. We used PHP 5 for server side scripting and MySQL 5.6 for data storage. The server was their existing Windows 2012 server.

How our new solution has performed?

We took precise care while analyzing the requirements and developing each modules. The only intention was to make system work faster, better reporting, simultaneous user transactions, faster search - data fetching and desktop like user experience on web application using keyboard shortcuts.

Our new solution seems to have worked nice and have improved their work flow. The client has posted an image showing our success!



Existing business application & problems

The administration team has different locations around the campus. They have different departments like stock keeping is done from the warehouse and accounting from main administration office. These departments were connected via internal network and internet if someone managing the data remotely.

They have developed their own accounting system on top of Microsoft Access and using it since 1995. The system was good fit for earlier days were team size is relatively small and system is lighter on data storage part. The system was divided into two functional layers one for front-end user interaction part and the other was database part.

They start facing the problem as their database starts growing, team start expanding and requirement of simultaneous user transactions. Access is not a RDBS fundamentally so it lake of modern age database features like indexing, key relationships, views creation, full-text search and modern performance tuning.

As their data grew significantly the module load time were getting higher. Some modules were taking minutes to load for user input. They were seeing huge delay when the application database on other department server and user is connected via local network. Finding records were taking more time as Access need to search though huge bulk of records. Reporting were their most used module and it was taking more time to pull data and run reporting business logic on it to prepare the report.

So when they contact us for new system development they have a condition that all their existing data should be migrated to new solution. They gave us complete walkthrough of their existing system. Then give us overview of all the problems they were facing.

How we approach the problem and solution we developed

We started analyzing their requirement. We used their current application on our server to understand it better. We asked them about use-cases and business process flow. We listened and logged every piece of information they provided and the request for new/extra features they need.

We targeted their use-case and problems. And based on our analysis and discussion with client we found that they need application that covers below points

- Central Database
- Open Source-technology
- Simultaneous multiple user transaction
- Easy and intuitive front-end user experience
- Easy deployment
- Faster data-processing
- Customized, easy and printable reports

Now it was our task to choose the right framework and software stack that satisfy the mentioned requirements.

We choose Web Application in this case. We may have gone for Windows Desktop application as they were already using Windows Server 2012. Considering reporting, heavy usage of charts, portability, device independence and OS independent we decided to go for web application path.

Based on our research we have think of dividing entire application into 3 distinct ways

- Business Logic and Data storage layer
- Web Front-end layer
- Service layer for asynchronous requests

Development process:

The development process is divided into 3 sections in altogether:

i. Analysis and requirement finalization
ii. Project management
iii. Communication
iv. Design
v. Technical details

Analysis and requirement finalization:

We took enough time analyzing their requirements. There were some areas which were use-case and organization specific. We asked for use-cases to know how they operate modules and how they would like those modules to behave on new application.

Project management:

After requirement gathering and features finalization we started working on system. We choose agile development methodology. We divided the system into set of modules and started working on core (most independent) module to specific (most dependent) module. We used trello for managing our project. We invited our client on board so they can see the work progress in real time and provide feedback in real-time manner.

Project management is one of the core fundamental elements that makes development efficient, puts us and our client on same page, open system where client can see team working project and each member working on module specific tasks. As we develop and test module. We ask our client to finally test the module against the real-world case. After testing he can directly notify the team-member about the update.

Communication:

Analysis is the part where communication is heavily used. Once analysis is completed we used to talk to our client when new module developed and we need to show case the update in real-time screen-share basis. Or times when client wants to make some part of application clear to us. The project management is efficient enough that client needed to talk to us once or twice while development life cycle for roughly 8 months.

Design

Design is application consumer’s first and foremost important as they would spend more time working with front-end of system. We used Boostrap based responsive design that looks uniform across devices. The design theme was HTML5 and CSS3 based. Their concern was the web application should behave the same way as desktop app. To achieve we heavily used jQuery for front-end engineering.

Technical Specification

As we stated we choose the web application path for the central database, easy deployment and device and OS independence it offers. We choose the opensource technology stack that would not require our client to purchase additional licenses. We selected PHP for server scripting and MySQL for database management.

Their primary intention was the new web application should work fast. So we designed the proper relational schema on MySQL and indexed the required entities. We designed the code that mix of asynchronous and synchronous requests to server for balance the load and faster data-processing.

We utilize Git for version management. As we are working in team it is essential to use version management for better collaboration. Release branch manages the upmost version of release ready code. We maintain the same version on production server and GIT production branch.